A Project Report

On

“iNotebook”

Submitted By

Mr. Tanmay A. Naik

Submitted To

NaranLala College of Professional and Applied Sciences,
Veer Narmad South Gujarat University, Surat.

Year: 2023-2024

NARANLALA
COLLEGE OF PROFESSIONAL & APPLIED SCIENCES
BHAGVATI SANKUL, NEAR ERU CHAR RASTA,
NAVSARI – 396 450

CERTIFICATE

This is to certify that **Mr. Tanmay A. Naik,** Exam No.1468 student of **B.C.A. 5th semester** of our college have successfully prepared and submitted Project Report on "iNotebook" as a partial fulfillment for the course of **Bachelor of Computer Application (Minor Project Sem. V)** during the academic year **2023-2024**.

Date: 04-10-2023 Guide: Dr. AB. Patel

Acknowledgment

Dear Reader,

I sincerely feel that I am not the only one to be credited for this Web Application. This mini project is an integrated effort of all self-study from Online Forums, YouTube tutorials, Official Docs, and all those concerned, by whose co-operation and practical guidance; I achieved its completion.

It is a curriculum to undergo project work in any system. I preferred the **"iNotebook"** at NaranLala College of Professional and Applied Sciences, Navsari.

This project helped me enhancing my skills in overall aspects becoming a web developer or an IT professional. Solving the errors without external support was a part of learning how to solve our problems and not to be dependent on anyone anytime.

I would like to take this opportunity to express my sincere gratitude to Principal Dr**. Sunil M. Naik,** Head of Department (BCA) **Dr. Ashish B. Patel**, **and** the faculty members of the Computer Science department.

Yours truly,

TANMAY NAIK

Abstract

The report reflects the journey of making a Full Stack MERN Application. It incorporates my learnings from errors, opening it for open source, hosting it on the web, and working with dynamic web technologies and libraries. I have gained knowledge of not only programming but about hosting on the web, finding vulnerabilities, SEO, designing paradigms, typography, and working with DB on the cloud.

I would like to thank my 2 FPG's (Friend, Philosopher, Guide):

1) Dr. Ashish B. Patel (who stated me to enhance more possible features in my web application and guide me throughout the project development).

2) Abhay Naik (Project Manager Inventyv Software Services PVT LTD, Who helped me learn React JS, and MERN and keeps me updated about what tech stack I learn next).

ABOUT

iNotebook

Welcome to the most convenient way to keep track of your thoughts, tasks, and ideas – **iNotebook**

iNotebook is designed with simplicity and productivity in mind.

Here's what sets us apart:
User-Friendly: We've created an intuitive and easy-to-use interface so you can start jotting down notes right away, without a learning curve.

Accessibility: Access your notes from any device with an internet connection. Whether you're at home, work, or on the go, your notes are always at your fingertips.

Thoughts: Categorize your notes with tags, folders, or labels, making it simple to find what you need when you need it.

Secure: Your data is important to us. We prioritize your privacy and security, ensuring your notes are kept safe.
Customizable: Personalize your notes with different fonts, colors, and formatting options to make them truly yours.

Get Started Today Join thousands of users who have already simplified their note-taking experience.

Have questions or suggestions? We'd love to hear from you! Contact our support team at:

findtanmay10@gmail.com
(+91)7096754251

Thank you for choosing **iNotebook**!

CONTENTS

Abbreviations	
API	Application programming Interface
BSON	Binary JavaScript Object Notation
CLI	Command Line Interface
HTTP	Hypertext Transfer Protocol
JSON	JavaScript Object Notation
JSX	JavaScript XML
MVC	Model View Controller
MERN	MongoDB, Express, React JS, Node JS
NPM	Node Package Manager
NoSQL	Not Only Structured Query Language
RDBMS	Relational Database Management System
SPA	Single Page Application

Introduction

The COVID-19 crisis has affected societies and economies around the globe and will permanently reshape our world as it continues to unfold. While the fallout from the crisis is both amplifying familiar risks and creating new ones, change at this scale also creates new openings for managing systemic challenges, and ways to build back better.

The most important sector That Got Affected Due To covid is education Sector, Due to online lectures and google classroom the students needs to save their notes locally and all scrambled

I personally face the same problem that's when I got opportunity to enhance my coding skill too and the coding notes were all scrambled I twisted both the problems and the idea that came was iNotebook.

"iNotebook – Your Notebook on The Cloud" is an opensource web Applications helping both students and teachers to manage their notes and access whenever needed.

Project Details:

●This web application delivers students an one stop solution where they can get their course notes whenever they need.

●This web application will be a boon for the students as well as teachers who want to save/manage new/current notes.

●Teachers can manage notes based upon their subjects in different classes.

●After successful registration, teachers and students can access their respective notes whenever need.

Purpose

●Provides Notes management in more efficient way.

●Access current Notes.

●Reduce the cost of the offline and tedious process of Notes making and Photo copies.

●Faculty can manage their Notes according to the module.

Project Scope

●Perfect for Institutions to save 1000s of notes.

●Register details in Mongodb, accessible only to the app developer (Excluding Password).

●Teachers can focus more on teaching instead of worrying about data handling.

●Async requests through API, MongoDB Compass.

Limitations:

- Not included restrictions of file size.
- Not included Virtual Meeting Feature (like Zoom).

System Requirements

1.Hardware Requirements:

- If you want to deploy the app in the local environment:
- PC/ Laptop.
- Minimum 4 GB RAM.
- Processor 1.4 GHz 32/64-bit.
- Internet connection.

2.Software Requirements:

- Windows 8.1(minimum)
- Latest Node JS, & NPM installed.
- Internet connection.
- Browser(required).

System Analysis
Requirement Analysis

MERN Stack:

1) MongoDB: A free, open-source, cross-platform, document-oriented cloud database. Developed while keeping scalability and developer agility keeping in mind. Instead of storing data in rows and columns, it stores JSON documents in collections with dynamic schemas making it easier to store and combine data of any structure, without complex validation rules, and schemas, with flexible data access, and rich indexing functionality.

2) Express JS: Web application framework that runs backend applications. Express runs as a module in Node JS, it can handle the routing of requests to the right parts of the application.

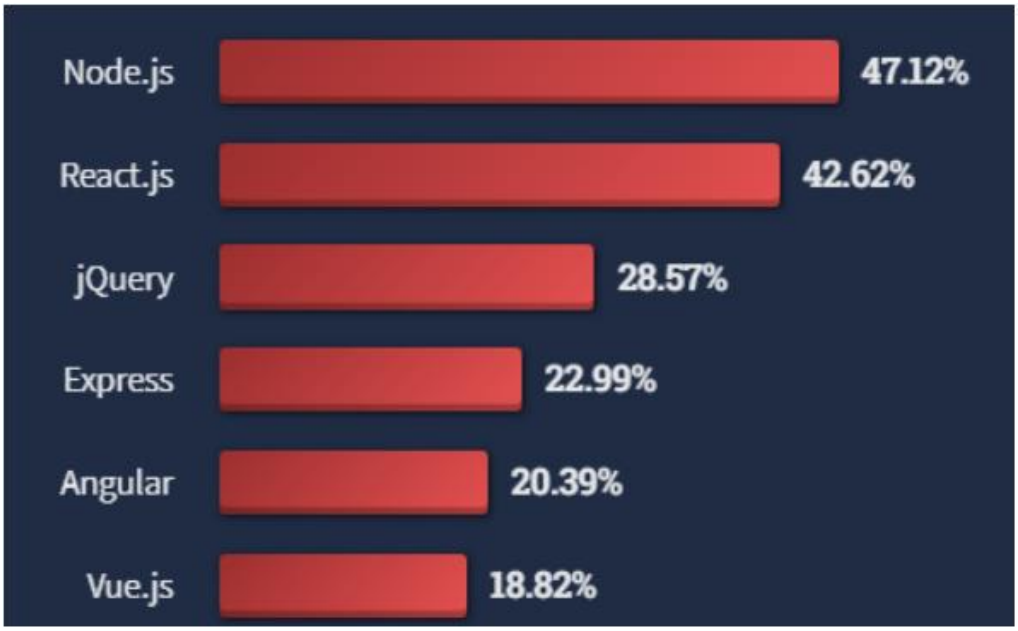
We use Express to perform two functions:

A) Send the front-end to the remote browser when a user browses our app.

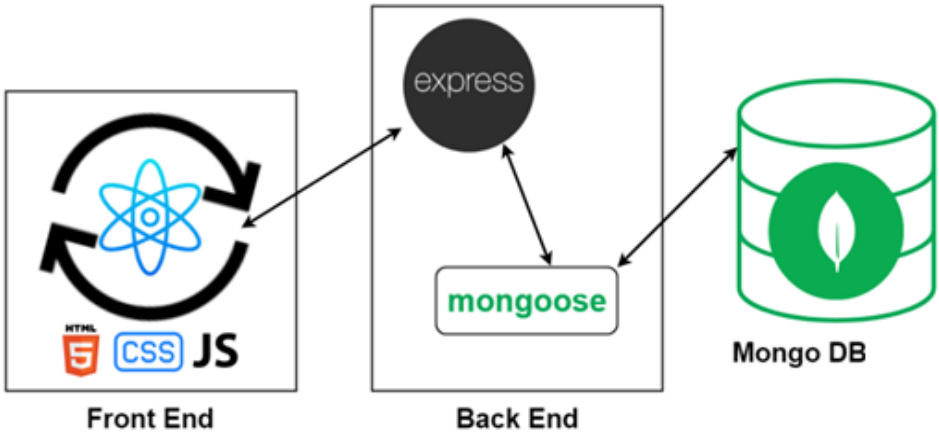
B) Provide REST API that the front-end can access using HTTP network calls to access the database.

3) React JS: JavaScript library developed by Facebook to build interactive UI. React breaks the front end into components. Each component can hold its state that the parent can pass down to its child components, and the child can pass changes back to the parent component using callback functions.

4) Node JS: JavaScript runtime environment that runs our back-end application(via Express). Node JS is made upon Google's V8 JS engine, used in the Chrome browser. Third-party modules are installed using npm(Node Package Manager). Node JS is an asynchronous, event-driven engine where the application makes a request and then continues working on other useful tasks rather than stalling until a response. The application receives results via a callback, ensuring a lot of operations are performed in parallel when scaling applications.



Node JS Usage 2022.



3 Tier MERN Architecture

Goals for conducting Requirement Analysis:

-Identify customers' needs.

-Evaluate the system for feasibility.

-Economic and Technical analysis.

-Functions to system elements.

-Enforcing constraints.

Steps included in Requirement Analysis:

-Problem recognition

-Evaluation and synthesis.

-Specification

-Review.

Requirement Types:

1)Functional Requirements:

Involves Input Output processes error handling.

2)Non-functional Requirements:

1. Physical environment
2. Interface, User, human factors, Documents, &Quality assurance.

3) Validation:

Are the requirements complete, accurate, uniform, and verifiable? iNotebook has fulfilled all the above requirements. It is accessible via any electronic device that has a browser on it. It fulfills the requirement of helping instructors create and manage new notes whose learners can learn from anywhere. The system runs successfully without errors.

We can proceed further to the FAST technique:

FAST(Facilitated Application Specification Techniques):

The Objectives of FAST:

-Specify requirements.

-Identify the problem.

-Propose a solution.

-Negotiate between approaches.

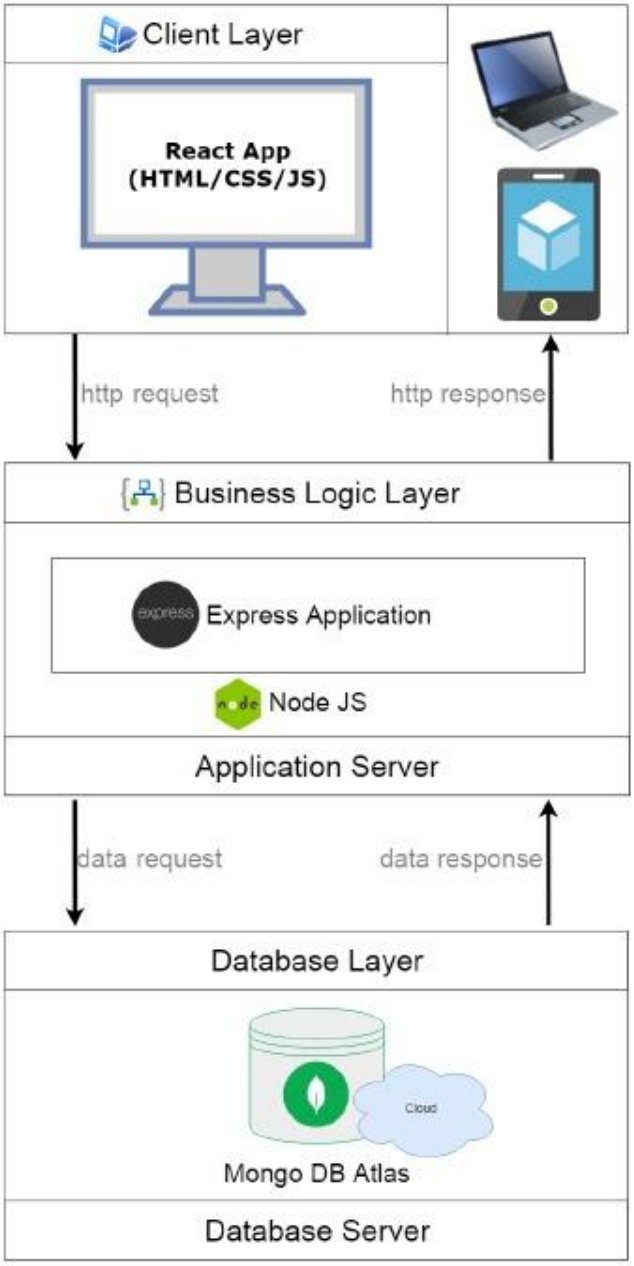
Feasibility Studies

iNotebook is feasible, it does the task for which it was designed, as a learner I have incorporated all the simple tasks that are performed. As an open-source project, I foresee including many amenities inside it as time permits. Operational and Technical feasibility is covered as it's secured, cloud-based, and hosted on the web, so serves its purpose.

As an open-source project, economic feasibility is already satisfied and the product is free to use. Schedule feasibility is also kept in mind as the testing environment is hosted on a dummy site and the actual production build is hosted on the main site.

Main Module of System

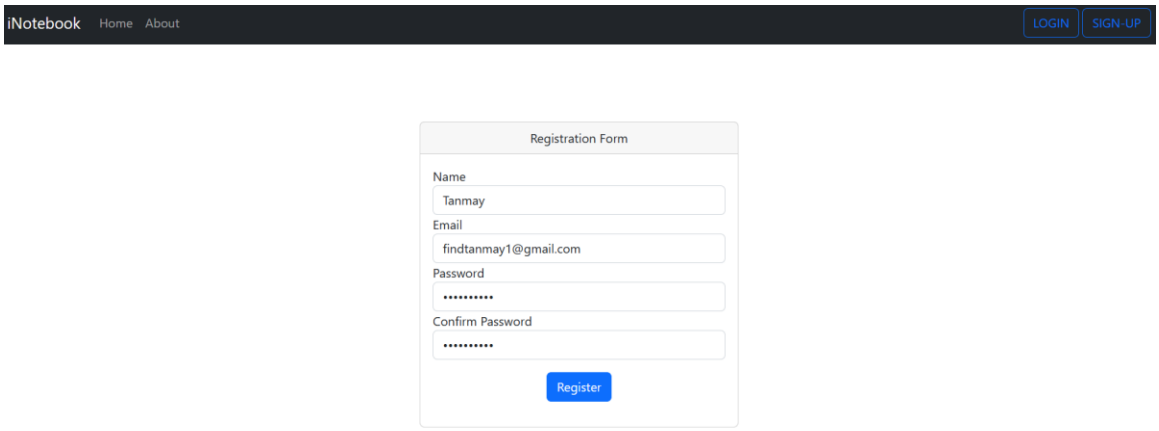
Main Module of System Overview:



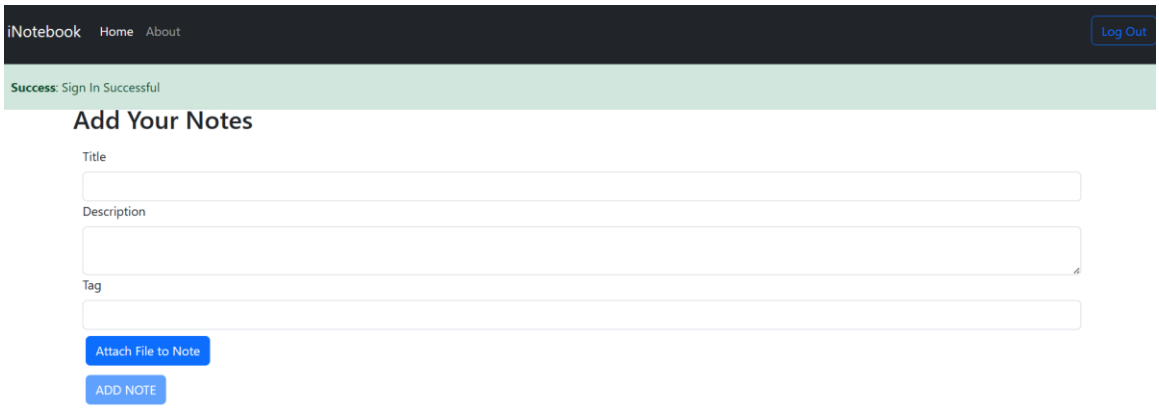
User Privileges

User Registration

REGISTRATION FORM:



POST SIGN-IN:

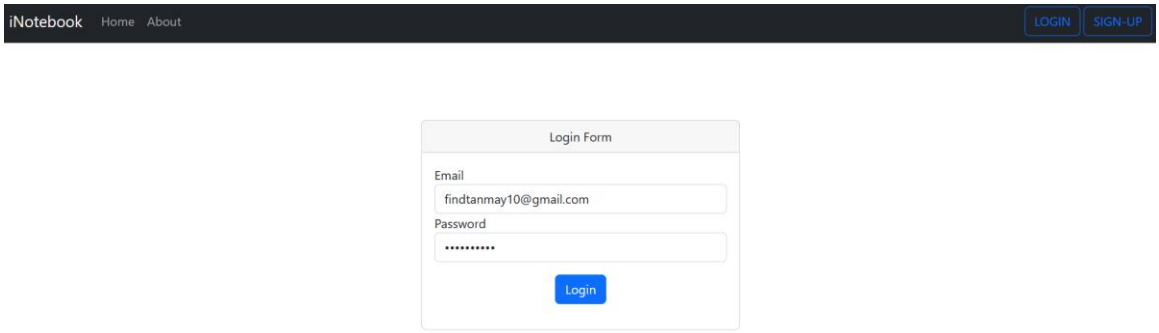


Notes (CRUD Operations)

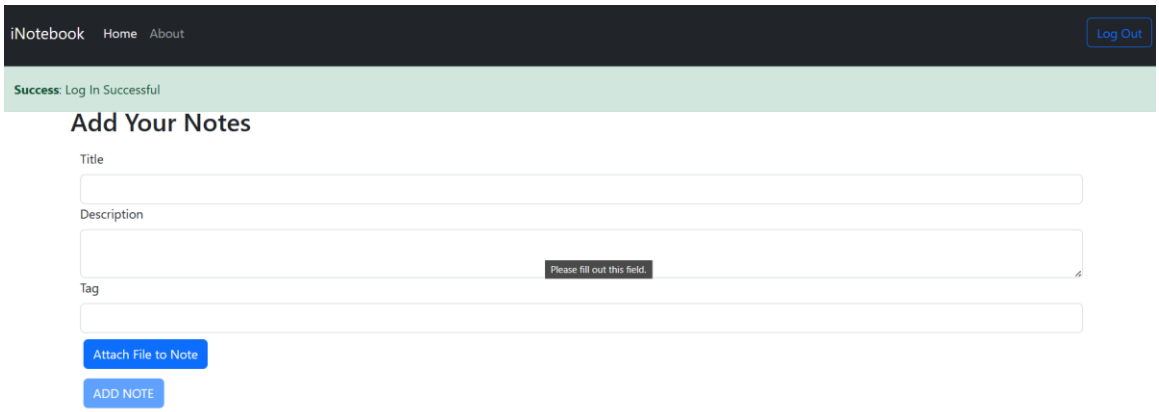
CREATING NOTES:

The User can create Notes by visiting and successfully logging in:

LOGIN PAGE:

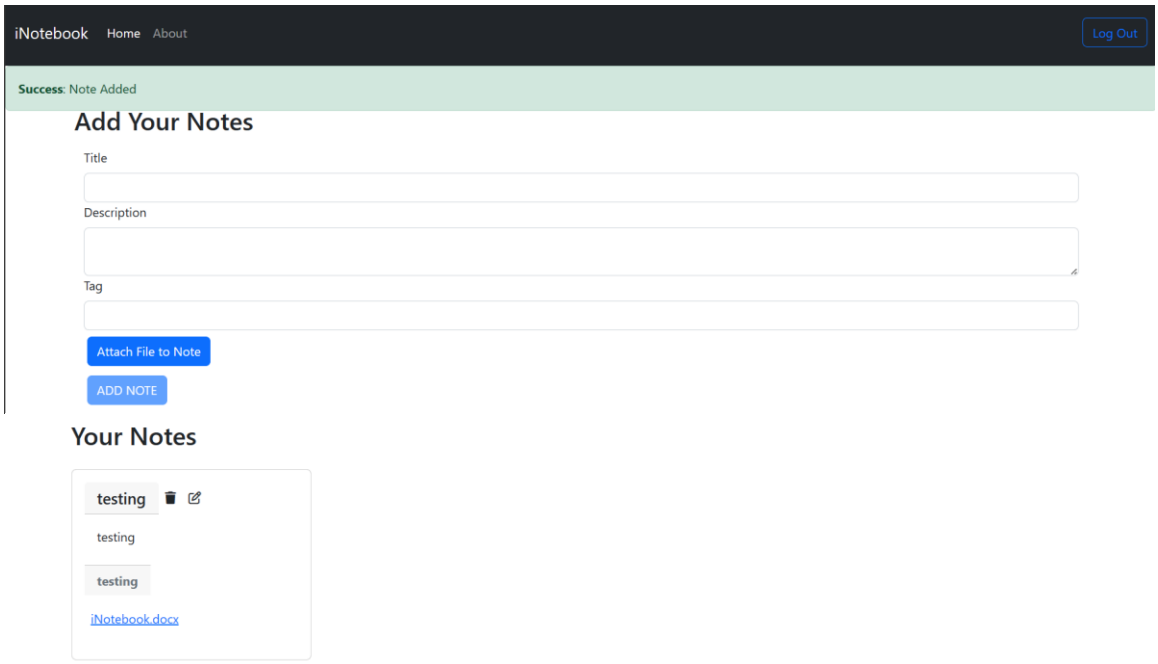


POST LOGIN:



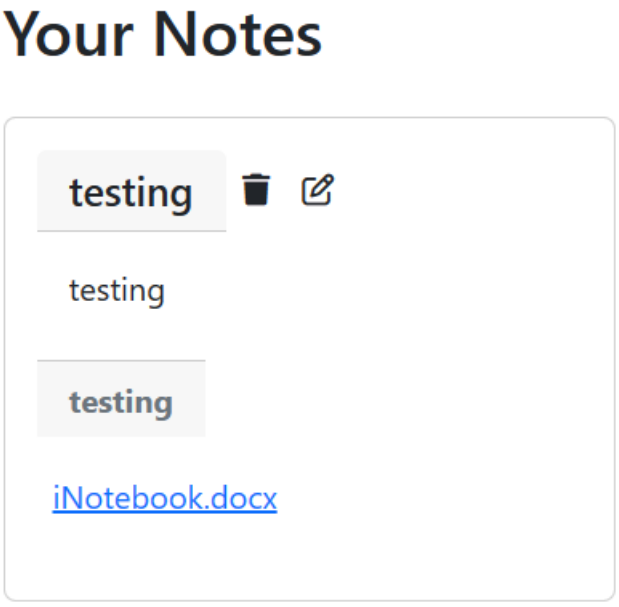
CREATING NOTES:

Creating a Note will insert the content into MongoDB Compass and asynchronously append the new Note at the end of the notes page.



READING NOTES:

Reading notes is carried out by fetching notes data from Mongodb Compass asynchronously.

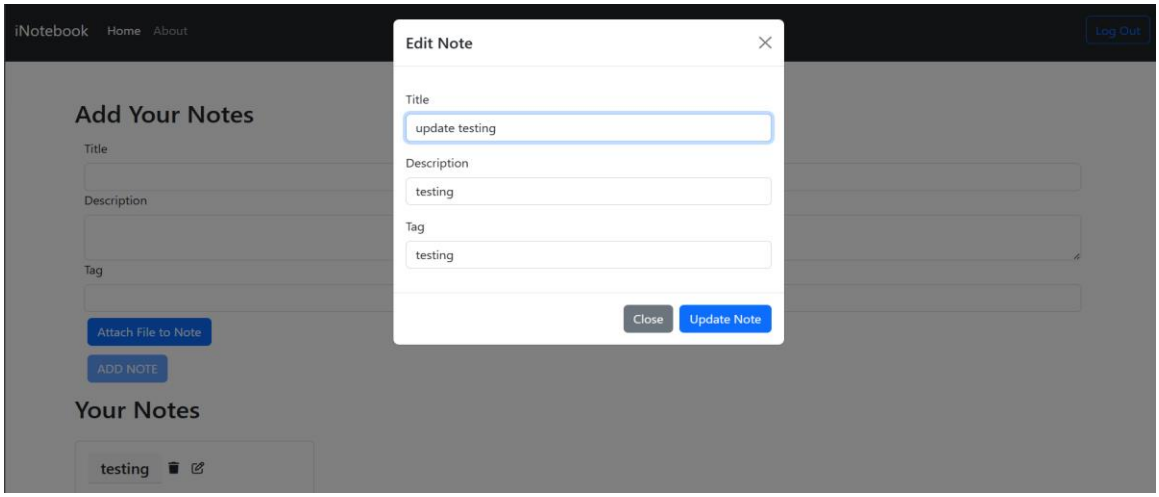


UPDATING NOTES

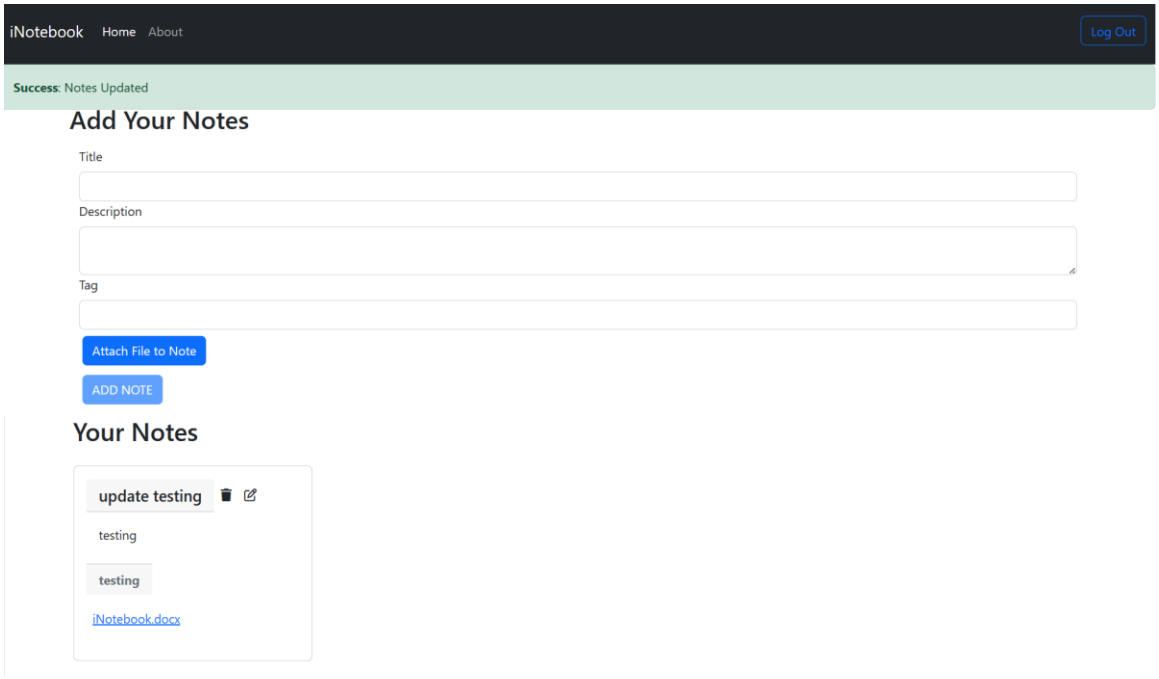
Updating a particular note will only update the note with a particular ID in MongoDB compass at the same time it will update the same note at display part.

Update modal will be launched at the onClick() event set on the font-Awesome icon pen-paper.

UPDATE NOTE MODAL:

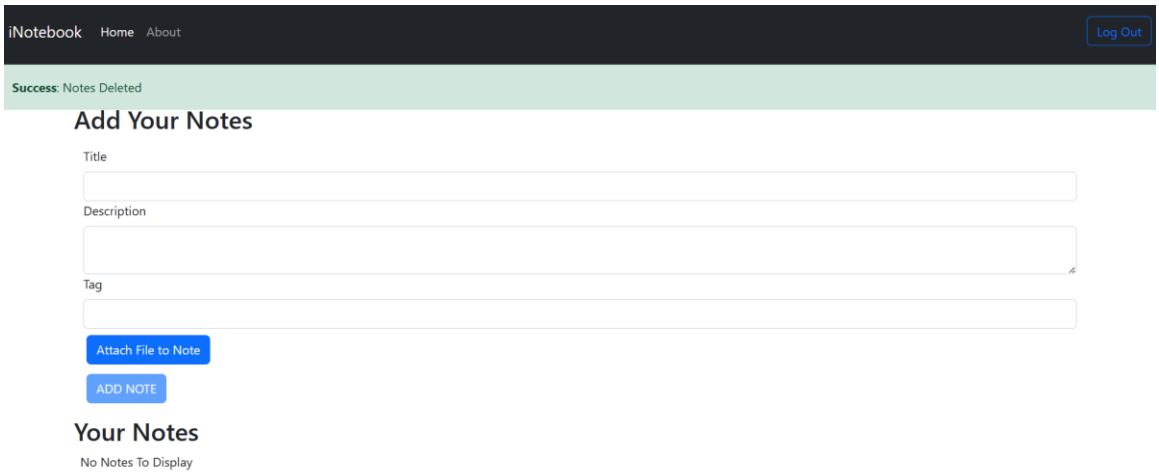


POST-UPDATE NOTE:



DELETING NOTES:

Deleting a note will permanently remove all details of the particular note in MongoDB Compass at the same time it will remove that note from the notes that are being displayed.



Tools and Technologies

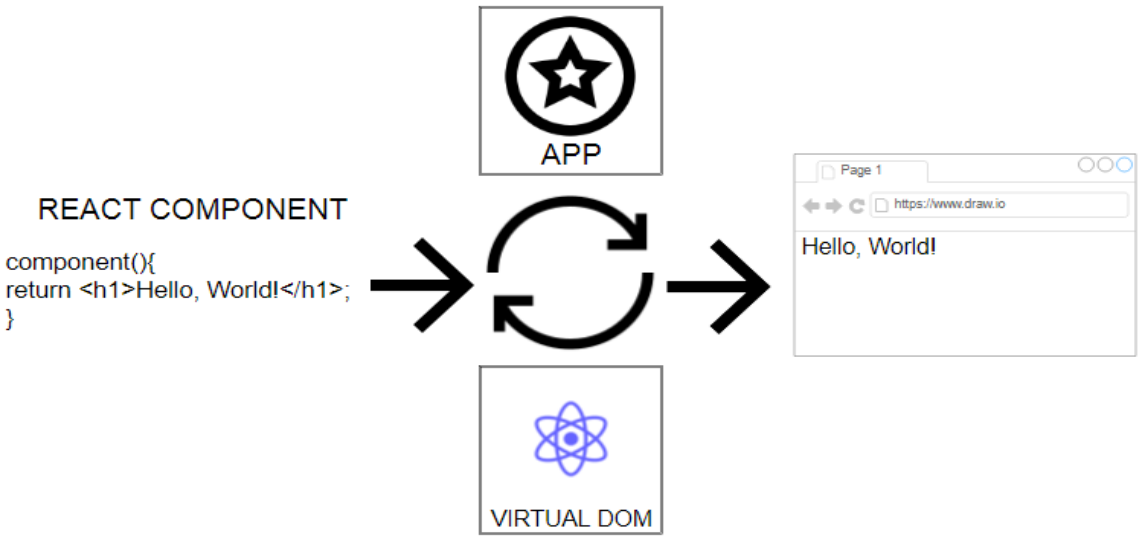
Overview

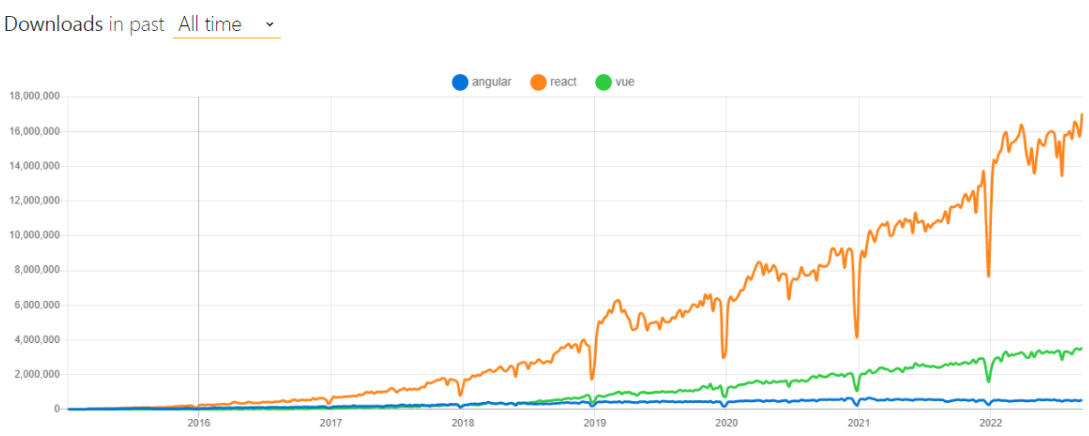
iNotebook - is a full stack MERN (MongoDB, Express, React, NodeJS) Application.

Whatever tech stack that's developed using NodeJS and is currently running on the web requires node modules, node modules are a group of programs packed in one program. One module can depend upon other 1000s of modules, another module can use the module at the same time.

Why React JS?

- ➔ Large developer community With 195k Stars on Github
- ➔ Everything is component
- ➔ Easy to learn compared to Angular JS
- ➔ Reuse of components
- ➔ Efficient debugging
- ➔ Routing Feature(loads pages with getting refreshed)
- ➔ Virtual DOM
- ➔ Effective Single Page Application
- ➔ JSX to increase performance
- ➔ Unidirectional data flow
- ➔ Easily develop dynamic websites
- ➔ Open Source





Popularity of React all time among Angular, React and Vue

CSS and Bootstrap

CSS is written from scratch keeping in mind the design rules, typography, font ligature, and responsiveness in mind. Total lines of CSS exceed 800 lines including inline(where it was the need of hour).

Bootstrap is used for modals, react version of Bootstrap: react-bootstrap and react-bootstrap-carousel modules were used to combine with inline JSX.


App.css
src > App.css > .App
.App {
text-align: center;
}
.App-logo {
height: 40vmin;
pointer-events: none;
}
@media (prefers-reduced-motion: no-preference) {
.App-logo {
animation: App-logo-spin infinite 20s linear;
}
}
.App-header {
background-color: #282c34;
min-height: 100vh;
display: flex;
flex-direction: column;
align-items: center;
justify-content: center;
font-size: calc(10px + 2vmin);
color: white;
}

Node Modules

Backend:

1. **Cors:** Cross-Origin Resource Sharing allows a server to indicate any origins (domain, scheme, or port) other than its own from which a browser should permit loading resources.

2. E**xpress:** the primary use of Express is to provide server-side logic for web and mobile applications.

3. **Mongoose:** Mongoose is a MongoDB object modeling tool designed to work in an asynchronous environment. Mongoose supports both promises and callbacks.

4. **Path:** provides a way of working with directories and file paths.

Frontend:

1. **jquery, react-bootstrap:** common frontend styling and functionalities.

2. **react-bootstrap-modal:** for modal in React.

3. **react-icons:** to add icons in our app.

4. **react-dom:** an efficient way of managing DOM elements of web page.

5. **Font-awesome:** for guileful hiding of delete and update button inside icon.

MongoDB Compass

It’s a MongoDB GUI.

To get started we can:

1 - Visit mongodb.com -> Docs -> Mongodb Compass

2 - Download -> Launch

3 - Connect

4 - Copy Connection String

5 - Paste the string to your API call

DATABASE SCHEMA:

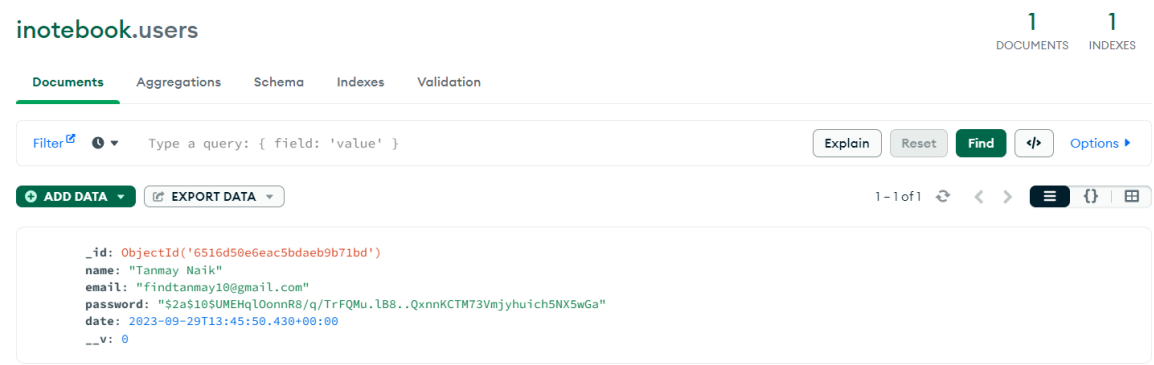
1)Notes:

```
backend > models > Note.js > ...
1   const mongoose = require('mongoose');
2   const { Schema } = mongoose;
3
4
5   const NotesSchema = new Schema({
6     user: {
7       type: mongoose.Schema.Types.ObjectId,
8       ref: 'user'
9     },
10    title: {
11      type: String,
12      required: true
13    },
14    filePath: {
15      type: String
16    },
17    fileName: {
18      type: String
19    },
20    description: {
21      type: String,
22      required: true
23    },
24    tag: {
25      type: String,
26      default: "General"
27    },
28    date: {
29      type: Date,
30      default: Date.now
31    }
32  });
33
34  module.exports = mongoose.model('notes', NotesSchema);
```


inotebook.notes
1
DOCUMENTS
1
INDEXES
Documents
Aggregations
Schema
Indexes
Validation
Filter
Type a query: { field: 'value' }
Explain
Reset
Find
Options
ADD DATA
EXPORT DATA
1 - 1 of 1
_id: ObjectId('651ad69ed80bd6f3b645f1d0')
user: ObjectId('6516d50e6eac5bdaeb9b71bd')
title: "testing"
filePath: "iNotebook.docx"
fileName: "uploadfiles\-QuA2IwRfYl4BUsTZGzfeKRd.docx"
description: "testing"
tag: "testing"
date: 2023-10-02T14:41:34.292+00:00
__v: 0

2)user:

```
backend > models > JS User.js > ...
1   const mongoose = require('mongoose');
2   const { Schema } = mongoose;
3
4   const UserSchema = new Schema({
5     name: {
6       type: String,
7       required: true
8     },
9     email: {
10      type: String,
11      required: true,
12      Unique: true
13    },
14    password: {
15      type: String,
16      required: true
17    },
18    date: {
19      type: Date,
20      default: Date.now
21    }
22  });
23  const User = mongoose.model('user', UserSchema);
24  module.exports = User;
```



API

Representational State Transfer Application Programming Interface is a tech that uses HTTP requests to access and use data, The data can be used to carry out GET(Read), PUT(Update), POST(Insert), and DELETE(Delete) operations against concerning data resources.

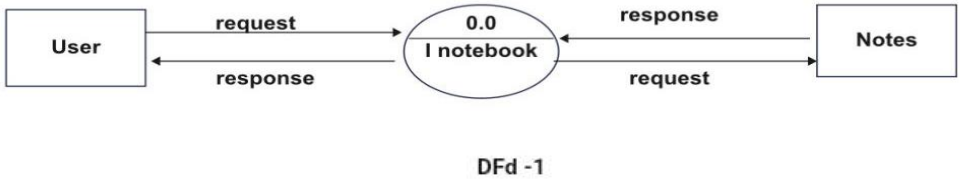
METHOD	ROUTE	DESCRIPTION
GET	router.get(“/fetchallnotes”)	GET ALL NOTES
PUT	router.put(“/updatenote/:id”)	UPDATE NOTE
POST	router.post(“/addnote”)	CREATE NOTES
DELETE	router.delete(“/deletenote/:id”)	DELETE NOTES

System Diagrams

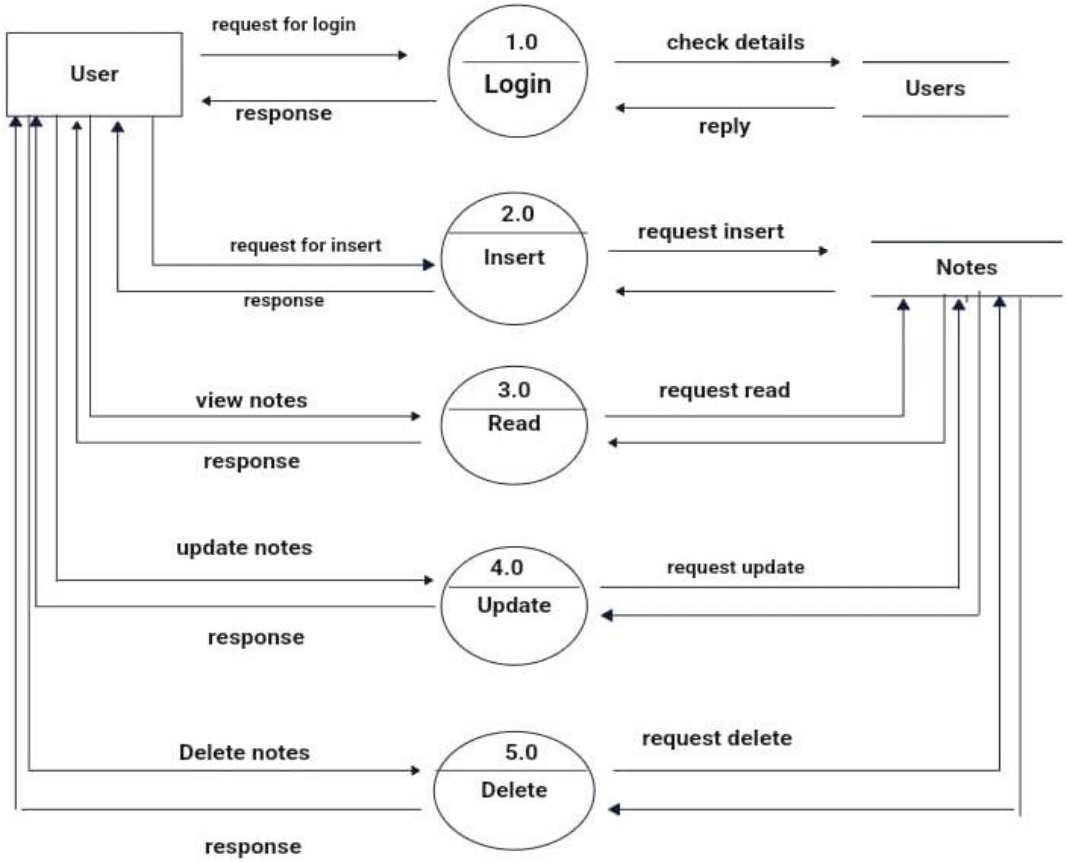
Data flow diagram:

The flow of data of a system or process is represented by DFD. The DFD also provides information about the outputs and inputs of each entity and the process itself. A data-flow diagram has no control flow there are no decision rules and no loops.

Context level
0-level



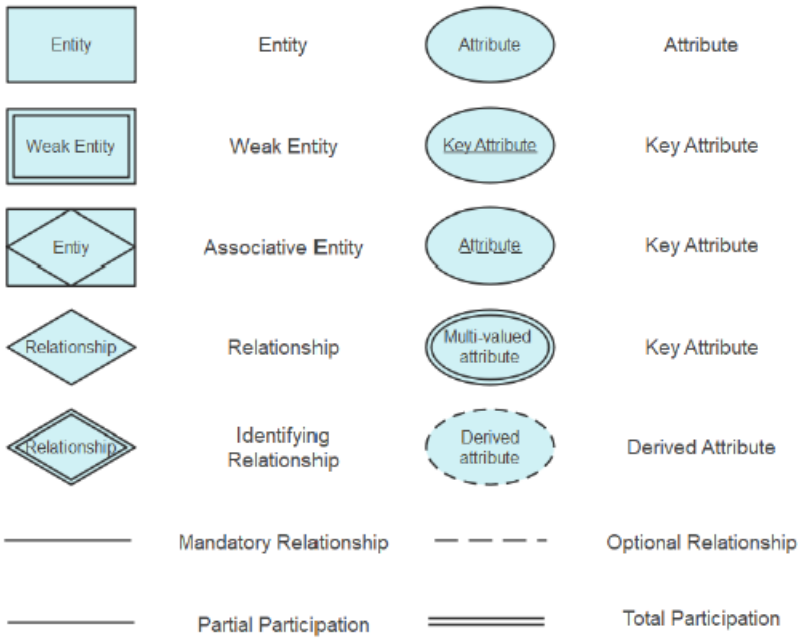
DFd -1

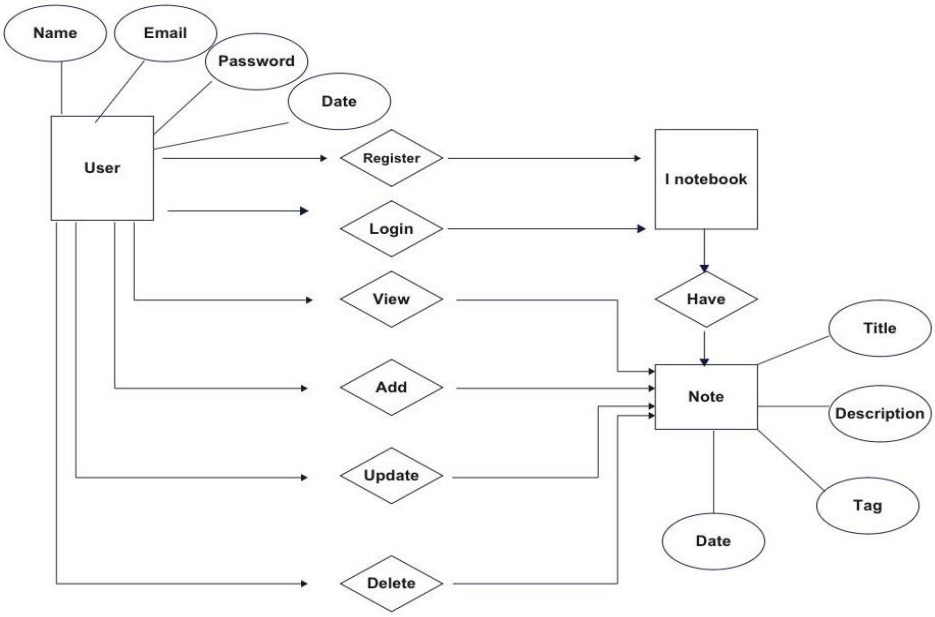


Entity Relation Diagram:

An entity-relationship model (or ER model) describes interrelated things of interest in a specific domain of knowledge.

A basic ER model is composed of entity types (which classify the things of interest) and specifies relationships that can exist between entities (instances of those entity types).




Name
Email
Password
Date
User
Register
Login
View
Add
Update
Delete
I notebook
Have
Note
Title
Description
Tag
Date

Scope of Further Development

New features that can be added to this application:

- New User Registration, Log In user Interface.
- Domain upgradation.
- Adding Web Notifications.
- Adding paid Reminder Notes with a payment interface.
- Adding in-built lecture scheduler tool (For Teachers), updating schema for inserting videos Notes.
- The above features will be added by me as I learn and develop new skills, and add them to this application as time permits.

REFERENCES:

React - https://react.dev/

Bootstrap - https://getbootstrap.com/

Npm - https://www.npmjs.com/

MongoDB - https://www.mongodb.com/

Font-Awesome - https://fontawesome.com/

Code-with-harry - https://www.codewithharry.com/

Youtube: https://www.youtube.com/@CodeWithHarry